XII Международная научно-практическая конференция студентов, аспирантов и молодых учёных
«Молодёжь и современные информационные технологии»

ВИДЫ NOSQL И ИХ СРАВНЕНИЕ С РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ

Ю.В. Курцев, Г.П. Цапко
Национальный Исследовательский Томский политехнический университет
jurykurtsev@gmail.com

Введение

В последнее время в связи с развитием Интернета активно развиваются поисковые системы, социальные сети и высоконагруженные сервисы, которые должны обрабатывать большие массивы информации и отвечать на огромное число запросов. Это требует не только максимального учета специфики обрабатываемой информации, но и перехода на распределенные вычисления. Никакой сколь угодно мощный сервер в не способен в обеспечить нужную производительность.

Есть три основных требования к высоконагруженным приложениям[1]:

- Много данных: самые большие из веб-приложений обрабатывают объемы данных на порядки больше тех, что предполагались для управления реляционными базами данных;
- Огромное количество пользователей: исчисляются миллионами, доступ к системам одновременно и постоянно;
- Сложные данные: как правило, это приложения не простой обработки табличных данных, которые можно найти во многих коммерческих и бизнес-приложениях.

Технологии реляционных баз данных, которые доминировали в ИТ-индустрии с 1980 года, начали показывать свои слабые места при переходе к веб-масштабам именно в этих трех аспектах, поэтому все большее число людей начали искать альтернативу. Такой альтернативой стали NoSQL базы данных.

Основные черты

В отличие от реляционных баз данных, NoSQL не гарантирую выполнение требований ACID. Вместо этого Эриком Брюером, автором теоремы CAP, был предложен набор свойств BASE[2]:

- Базовая доступность (англ. basic availability)
- Гибкое состояние (англ. soft state)
- Согласованность в конечном счете (англ. eventual consistency)

Таким образом NoSQL жертвуют согласованностью ради доступности и устойчивости к разделению.

Другие характерные черты NoSQL:

- Отсутствие фиксированной схемы данных;
- Линейная масштабируемость;
- Отсутствие стандартного языка запросов;
- Хорошая горизонтальная масштабируемость;
- Различные типы хранилищ.

Виды NoSQL баз данных

Выделяют 5 видов NoSQL баз данных[3].

Хранилища "ключ-значение"

Простейшей формой хранилища системы NoSQL является хранилище пар ключ-значение. Каждый ключ ставится в соответствие значению, в форме произвольных данных. СУБД не располагает информацией об этих данных.

Если разработчик использует структурированные форматы для хранения сложных структур данных, соответствующих ключу, он должен обрабатывать данные на уровне приложения: хранилище пар ключ-значение в общем случае не предоставляет механизмов для запросов ключей на основании некоторых свойств соответствующих им значений. Хранилища пар ключ-значение отличаются простотой их модели запросов, обычно состоящей из примитивов для установки, получения и удаления значений, но не предусматривают возможности добавления простых функций фильтрации на уровне базы данных.

Такие БД очень производительны, просты в обращении и легко масштабируются.

Примеры таких хранилищ — Berkeley DB, MemcacheDB, Redis, Riak, Amazon DynamoDB.

Хранилище семейств колонок

Основная идея колоночных СУБД — это хранение данных не по строкам, как это реализовано в реляционных СУБД, а по колонкам. Это означает, что с точки зрения клиента данные представлены как обычно в виде таблиц, но физически эти таблицы являются совокупностью колонок, каждая из которых по сути представляет собой таблицу из одного поля. При этом физически на диске значения одного поля хранятся последовательно друг за другом. Такая организация данных приводит к тому, что при выполнении запроса на чтение, в котором фигурируют только 3 поля из 50 полей таблицы, с диска физически будут прочитаны только 3 колонки. Это означает что нагрузка на канал ввода-вывода будет приблизительно в 50/3=17 раз меньше чем при выполнении такого же запроса в традиционной СУБД.

Кроме того, при колоночном хранении данных появляется возможность компрессии данные, так как в одной колонке таблицы данные как правило однотипные.

Недостатком колоночных СУБД является низка скорость выполнения операций на запись.

Документно-ориентированные СУБД

Документно-ориентированные СУБД – компьютерные системы, разработанные для хранения, получения и управления документно-

ориентированной или слабо структурированной информацией. В отличие от реляционных БД с их понятием Отношение (Таблица) эти системы оперируют абстрактным понятием Документ. Хотя каждая реализация документно-ориентированной БД определяет это понятие по-разному, в общем, все они подразумевают инкапсуляцию и кодирование сохраняемой информации в некотором стандартном формате. Например, в XML, YAML, JSON, BSON, а также в бинарных форматах, таких как PDF, документах Microsoft Office и т.п. Документы внутри документно-ориентированных БД некоторым образом похожи на записи или строки в реляционных БД, но являются более гибкими. Они не требуют наличия одних и тех же разделов, частей, ключей и т.п.

Документы адресуются в БД посредством уникального ключа, который представляет конкретный документ. Часто в роли такого ключа выступает обычная строка, путь к файлу и т.п. В любом случае, можно использовать этот ключ для получения документа из базы. При этом обычно СУБД строят индексы по таким ключам, так что получение документа из базы является весьма быстрым.

Примерами документно-ориентированных СУБД являются MongoDB, IBM Lotus Notes, CouchDB, Oracle NoSQL и др.

Граф-ориентированные СУБД

Граф-ориентированная база данных - база данных, основная модель хранения данных в которой — классический математический граф. Граф состоит из вершин и связей, где вершины представляют собой объекты данных, а связи - отношения между объектами.

В последнее время наблюдается бурный рост интереса к графовым БД в связи с тем, что такая система представления данных оказалась естественной и востребованной в современном мире различных социальных связей (Интернет, социальные сети и т. д.). К достоинствам графовых моделей БД по сравнению с традиционной реляционной моделью исследователи относят не только возможность естественной реализации графовых операций (поиска путей, выделения сообществ и т. п.), но и гибкую схему данных, позволяющую унифицировать хранение разнородных объектов.

Примерами графовых СУБД являются Neo4j, AllegroGraph, BigData, InfiniteGraph.

Объектно-ориентированные СУБД

Объектно-ориентированная база данных (ООБД) – база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов.

Объектно-ориентированные системы управления базами данных (ООСУБД) позволяют работать с объектами баз данных так же, как с объектами при программировании на объектно-ориентированном языке программирования. ООСУБД расширяет языки программирования, прозрачно вводя долговременные данные, управление параллелизмом, восстановление данных, ассоциированные запросы и другие возможности.

Некоторые объектно-ориентированные базы данных разработаны для плотного взаимодействия с такими объектно-ориентированными языками программирования, как Java, C#, C++, и т.п.; другие имеют свои собственные языки программирования. ООСУБД используют точно такую же модель, что и объектно-ориентированные языки программирования. Объектно-ориентированные базы данных обычно рекомендуется использовать в тех случаях, когда требуется высокопроизводительная обработка данных, имеющих сложную структуру.

Примерами ООСУБД являются Jasmine, ObjectDB, Caché, Matisse.

Сравнение NoSQL с реляционными СУБД

NoSQL имеют как преимущества так и недостатки перед реляционными СУБД[4].

Преимущества NoSQL перед реляционными СУБД:

- Широкий выбор типов хранилищ;
- Хорошая горизонтальная масштабируемость;
- Простота администрирования;
- Отсутствие жестко заданной схемы данных;
- Простой API для манипуляции данных.

Недостатки NoSQL перед реляционными СУБД:

- Отсутствие единого стандартного языка запросов;
- Большинство NoSQL баз данных не гарантирую выполнение требований ACID, что может повлечь потерю данных при отказе оборудования:
- Отсутствие поддержки целостности данных;
- Простой API для манипуляции данных затрудняет выполнение сложных выборок.

Литература

1. От SQL к NoSQL и обратно [Электронный ресурс]. - Режим доступа: http://www.osp.ru/os/2012/02/13014127/, свободный
2. NoSQL [Электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/NoSQL, свободный
3. Sadalage, Pramod J. NoSQL distilled : a brief guide to the emerging world of polyglot persistence / Pramod J Sadalage, Martin Fowler
4. Leavitt Neal. Will NoSQL Databases Live Up to Their Promise? // Computer. — 2010. — Vol. 43. — Pp. 12–14.